Application of Hypergraphs to SMCs Selection

Łukasz Stefanowicz, Marian Adamski, Remigiusz Wiśniewski, Jakub Lipiński

Application of Hypergraphs to SMCs Selection

Łukasz Stefanowicz[1], Marian Adamski[1], Remigiusz Wiśniewski[1], Jakub Lipiński[1],

[1] University of Zielona Góra, Institute of Computer Engineering and Electronics,
ul. Licealna 9, 65-417 Zielona Góra, Poland
{L.Stefanowicz, J.Lipinski}@weit.uz.zgora.pl,
{M.Adamski, R.Wisniewski}@iie.uz.zgora.pl

Abstract. The paper deals with selection of State Machine Components (SMCs) based on Hypergraphs theory. The entire selection process use Petri nets as benchmarks. As it is known, Petri nets are used for modeling of concurrency processes. The SMCs selection problem is classified as NP-Hard which means there does not exist polynomial algorithm which provides an exact solution. In the article we show three SMCs selection methods, advantages and disadvantages of each, results of comparison between traditional methods (exponential backtracking, polynomial greedy) and an exact transversal method based on hypergraphs theory, their efficiency and propriety. An exact transversal method allows to obtain exact solution in polynomial time if selection hypergraph belongs to xt-hypergraph class.

Keywords: Petri net, State Machine Component (SMC), hypergraph, exact transversal, concurrency hypergraph, sequential hypergraph, backtracking, greedy, algorithm of exact transversals.

1 Introduction

Petri nets are mainly applied in the modelling of concurrent processes since they enable describing both sequential and parallel relations [1], [2], [3]. Since the description mechanism of places and transitions is relatively simple, Petri nets are commonly applied. Unfortunately, the analysis may be impossible for the considerable sizes of the net and the exponential relation between its size and the number of states [1], [2], [4]. The selection of SM-components of a Petri net is applied in order to make the analysis possible, which helps to obtain a less complex problem [5].

The selection of SM-components may be performed with the use of classical methods, as well as with the use of the hypergraph model. It must be stated here that the application of hypergraphs allows the analysis in the polynomial time in the majority of cases [6], [7], [8].

In the article, three possible algorithms of covering have been presented, including the one which applies the hypergraph model: the backtracking algorithm, the greedy algorithm and the methods applying exact transversals of concurrency hypergraph [7], [9], [10]. The course of conduct consists in the determination of a concurrency hypergraph on the basis of the net. The hypergraph presented in a form of a matrix contains vertices (columns) which relate to places of a Petri net and also hyperedges (rows) reflecting relations between the places. Then a selection hypergraph is determined and subsequently an exact transversal, constituting the solution, is found [7], [8], [11]. The described process will be presented in details in the article.

One of goals of current PhD work is to develop algorithms using hypergraphs theory to analyze discrete states of concurrent automata. That goal implies following hypothesis: discrete states of concurrent automata can be efficiently and effectively analyzed using hypergraphs theory. Current article is a continuation of the work presented in [8].

2 Contribution to Collective Awareness Systems

The Collective Awareness Systems refer to algorithms that are considered as global systems. The most fruitful advantages are: easy diagnosis and usability. Industrial structures are constantly transformed. Infrastructure, technology resources, services and human potential permit the development of civilization and economy. Innovation is the main force of any economy, the essence is the constant search for new ideas and solutions. New technical solution should be based on Collective Awareness Systems because in this way it will be possible efficient use of facilities.

The algorithm and research methodology presented in the article are useful in case of global systems. Presented algorithms can be seen as a piece of a major system, which leads to the conclusion about the usefulness. Moreover, all methods shown in the article introduce new quality in the process of SM-Component selection. Reduction of the execution time of algorithms influences on financial savings. Proposed algorithms may be used as a part of the decomposition process of concurrent automata. Separate modules can be implemented on different FPGA devices, parts of the Collective Awareness Systems.

3 Main Definitions and Preliminary Notation

Current chapter contains the necessary definitions required to understand the topic.

Definition 1. Hypergraph H [12] is defined by a couple:

$$H=(V,E), \tag{1}$$

where:
$V=\{v_1,\ldots,v_n\}$ is a finite, non-empty set of vertices;
$E=\{E_1,\ldots,E_n\}$ is set of hyperedges, i.e., of set $P(V)$.

Definition 2. Graph G [12] is a particular case of hypergraph H. Formally a graph is defined by a couple:

$$G=(V,E), \tag{2}$$

where:
V={$v_1,\ldots,v_n$} is a finite, non-empty set of vertices;
E={$E_1,\ldots,E_n$} is a finite set of unordered pairs of vertices, called edges .

Definition 3. A Petri net [1], [2], [13] is a 4-tuple:

$$PN=(P,T,F,M_0), \tag{3}$$

where:
P is a finite set of places, T is a finite set of transitions,
$F \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs, M_0 is an initial marking.

Definition 4. A SM-Component PN' of a Petri net PN is such its consistent subnet:

$$PN'=(P,T,F), \tag{4}$$

that:

$$\forall t \in T' : |\cdot t| = |t \cdot| = 1 \tag{5}$$

$$P' = \cdot T' \cup T' \cdot \tag{6}$$

$$F' = (P' \times T') \cup (T' \times P') \cap F \tag{7}$$

Definition 5. A transversal T (hitting set, vertex cover) of hypergraph H is set:

$$T \subseteq V\ , \tag{8}$$

containing vertices incident to each of the hypergraph edges.

Definition 6. Exact transversal [10] D of a hypergraph H is a set:

$$D \subseteq V\ , \tag{9}$$

of vertices of hypergraph H, which is incident to all the edges of hypergraph H, while each edge is incident to exactly one vertex of set D which forms an exact transversal.

Definition 7. Exact transversal hypergraph H_{XT} (xt-hypergraph) is such a hypergraph of which all the minimum transversals are simultaneously exact ones [9].

4 Problem Formulation

The selection of SM-components is essential for the decomposition process [14]. It provides the base for the target SM-components which enable covering the whole Petri net. Approximate algorithms enable obtaining the result which is mainly an redundant one, and hence the result is not optimal [15], [16]. Exact algorithms (colouring, backtracking) enable obtaining results which do not contain redundant SM-components because a complete set of solutions is searched for, thus the obtained result is optimal and the number of SM-Components minimal. The analysis applying the hypergraph model [7] enables obtaining the exact result in a polynomial time for the vast majority of cases [8]. It should be stated here that the selection is a problem of the NP-hard class therefore the algorithm allowing obtaining an exact result in a polynomial time is not known. The selection process may be divided into three basic stages [7]:

- determination of the concurrency hypergraph on the basis of the elementary net,
- determination of SM-Components (SMCs),
- proper selection of SMCs and determination of the solution of the cover.

5 The State of the Art

The selection of SM-Components belongs to the class of NP-hard problems. It means that there is no universal polynomial algorithm allowing the optimal solution to be obtained in a polynomial time. The decomposition of a Petri net in the form of SM-Components selection may be carried out with the use of classical methods which apply graphs, as well as the theory of hypergraphs. The selection basing on the theory of graphs involves graph colouring [7], whereas the theory of hypergraphs applies the method of the determination of transversals of the exact hypergraph, proposed in [7]. The method has been expanded by the application of the exact transversals hypergraph, proposed in [8]. The exact methods demonstrate the exponential computational complexity [17], hence they require application methods. They, in turn, allow obtaining a solution comprising generally excessive/redundant subnets, in a acceptable time, though [7]. The method applying the exact hypergraph is not always effective since there is no polynomial algorithm allowing stating if a given hypergraph belongs to the class of exact hypergraphs. The method proposed in [8] involves the application of the hypergraph of exact transversals, and since there is a polynomial algorithm allowing testing if a given hypergraph belongs to the xt-hypergraph class in a polynomial time [10], it is possible to obtain an optimal solution on the condition that the selection hypergraph belongs to the xt-hypergraph class. In other words, it is possible to obtain exact cover, and hence an optimal solution provided that the selection hypergraph belongs to the class of exact transversals hypergraphs.

5.1 Backtracking Algorithm

The backtracking method enables obtaining the exact solution, and thus an optimal one, since it checks all the possible combinations. As a result, the smallest cover is found. The most considerable drawback of this solution is its computational complexity, which is exponential. It means that the method will work only with small nets. The algorithm's procedure is very simple. In the first step, a subnet is selected and then it is removed. If the remaining set of SM-Components still constitutes a net cover, the result is understood to be better than the previous one. In the subsequent stages, the algorithm searches for an optimal solution for all SM-Components of the elementary net. The backtracking method will always return the best solution, which is undoubtedly its immense advantage. Its biggest drawback is its exponential computational complexity, which in practice means that the given problem may not be solved, or the solution may not be determined in an acceptable time.

5.2 Greedy Algorithm

The greedy algorithm searches for the solution on the basis of a decision which is locally optimal [18], [19]. Each step of the method relies on the selection of a currently optimal solution. One of its advantage is its computational complexity. Its major drawback is the fact that the solution seen as locally optimal not always is globally optimal. Due to the fact, that the selection of SM-Components is equivalent to the hypergraph of selections, the greedy algorithm has been presented below in a vertex version. Its procedure may be presented as follows:

- At the beginning the cover set is empty.
- The selection of an essential vertex (the only one which belongs to a given edge). If such does not exist, a vertex of the largest degree is selected. Here is the moment when the selection of the locally optimal solution takes place. If the vertex is impossible to be determined univocally, the first possible one is selected.
- The selected, locally optimal, vertex is added to the cover set and removed from the selection hypergraph.
- The set is verified in terms of constituting a cover. If not, step 2 is repeated.

5.3 Algorithm of Exact Transversals

The algorithm of exact transversals has been described in [7]. The idea of the method has been presented in figure 1. The first step is the transposition of the sequential hypergraph matrix and the cyclical reduction called Cyclic-Core [4]. It involves the reduction of dominated columns and dominating rows. Then, for such an obtained hypergraph, the first exact transversal is determined. It is worth noticing that the whole process may be performed in a polynomial time provided that the hypergraph subjected to the cyclical reduction belongs to the class of exact hypergraphs. Moreover, the hypergraph should be verifiable in a polynomial time in terms of its affiliation to the class of xt-hypergraphs (an exact hypergraph is a generalisation of the xt-hypergraph), and hence yet at the stage of the hypergraph obtaining, after the reduction, it is possible to evaluate the probability of achieving the exact solution. As presented in [8], 80% of the tested nets contained a selection hypergraph belonging to xt-class. It means that for 80% of tested cases, it is possible to achieve an exact solution in a polynomial time, which undoubtedly is a great advantage of the method. The application of the theory of hypergraphs enables the reduction of the execution time from exponential to polynomial in the vast majority of cases.

6 Selection Process of SM-Components

The whole process of the selection of SM-Components may be presented in the following steps:

- An input/elementary net is given.
- On the basis of the input/elementary net, a concurrency hypergraph is determined in which vertices respond to the places of the Petri net, whereas hyperedges describe relations between the places.
- On the basis of the concurrency hypergraph, a sequential hypergraph is determined in which vertices respond to the places whereas hyperedges determine SM-Components [7].
- The proper selection of SM-Components with the use of the backtracking method, the greedy algorithm or exact transversals.
- Final solution is determined.

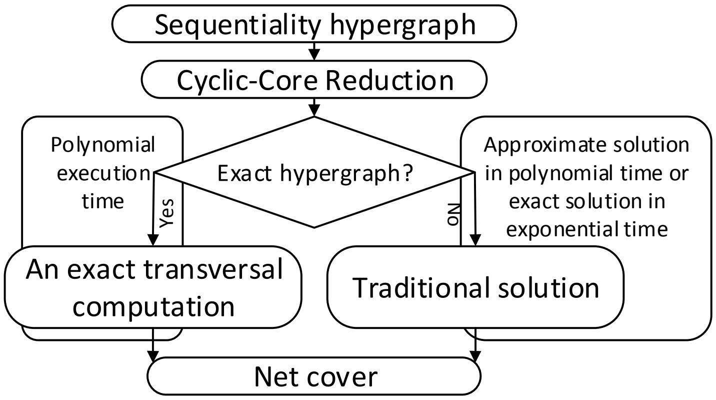
Sequentiality hypergraph
Cyclic-Core Reduction
Exact hypergraph?
Polynomial execution time
Yes
No
Approximate solution in polynomial time or exact solution in exponential time
An exact transversal computation
Traditional solution
Net cover

Fig.1. Idea of exact transversal method.

7 Experiments

The essence of the experiments is to use benchmarks to reveal the efficiency and effectiveness of the presented methods. The execution times for particular algorithms were compared with taking into account their sensitive moments and the results they return. Altogether, 30 benchmarks were used, nets of which belong to various classes:

MG (Marked Graph), FC (Free-Choice), EFC (Extended Free-Choice) as well as SN (Simple Net). The column “a minimal number of SMCs" included a division in terms of the applied algorithms: Backtracking (B), Greedy (G), Exact transversals (T).

For the ampleness of the data, only representative benchmarks were selected and are presented in table 1. As can be seen, the algorithm using exact transversals dominates over the backtracking and greedy ones in terms of the execution times, providing similarly good results. The greedy algorithm does not provide optimal value for MG_NP net, whereas the backtracking one takes a considerable time to be executed, unlike the exact transversal algorithm which in the shortest time returns the optimal solution. Both the exponential and the exact backtracking algorithm provide the optimal solution. The exact transversal method is as good as Backtracking which can obtain results in more than 200% shorter execution time. Presented results of research were obtained on the following workstation: AMD FX4100 CPU, 8GB RAM.

Table 1. Partial results of research.

Benchmark name, number of transitions, places	Cover determination time B [ms]	Selection determination time B [ms]	Cover determination time G [ms]	Selection determination Time G [ms]	Cover determination Time T [ms]	Selection determination Time T [ms]	Minimal number of SMCs B, G, T
bridge, 6, 8	119	338	128	254	2	162	2, 2, 2
cncrr002, 7, 11	281	504	288	449	2	162	5, 5, 5
Frame, 10, 13	204	748	201	387	2	188	3, 3, 3
P2N2, 7, 13	702	2901	711	1684	9	841	4, 4, 4
MG_NP, 5, 10	439	16521	438	644	5	188	3, 4, 3
Philoso2, 10, 14	382	651	312	577	2	194	6, 6, 6

8 Summary and Conclusions

The paper presents the selection of SM-Components of a Petri net applying the hypergraph model. It should be noted that the selection is an NP-hard problem. The main objective of the article was to present the possibilities of solving the selection problem with the use of hypergraphs. Thus, three algorithms were presented: the backtracking one, the greedy one and the one applying exact transversals. The last mentioned was proposed and described in [7], whereas its modified version was presented and subjected to research in [8].

The essence of the research was to demonstrate the effectiveness and efficiency of the particular algorithms. Due to the considerable sizes of research results, a group of representative benchmarks was selected and on their basis partial results were presented. It may be noted that the backtracking algorithm, which in fact provided optimal solution, takes frequently much more execution time. This is connected with the exponential computational complexity. The greedy algorithm provides the solution which is locally optimal, which not always is equivalent to a globally optimal solution. The execution time is frequently shorter than for the backtracking algorithm, but the solution may not be the best. The algorithm applying exact transversals provides exact solution for the vast majority of cases in a polynomial time [8], which is also confirmed by the presented research. It should be noted that the given solution is just as good as the one obtained with the use of the backtracking algorithm, which is characterised by an exponential computational complexity. The execution time for the algorithm of exact transversals is by over 30% lower than the execution time for the greedy algorithm and by over 200% lower than for the backtracking algorithm, the exact transversal method is as good as Backtracking.

References

1. Murata, T.: Petri nets: properties, analysis and applications. Proceedings of the IEEE 77 (April 1989) 541-580
2. Karatkevich, A.: Dynamic analysis of Petri net-based discrete systems. Springer (2007)
3. Bukowiec, A., Mróz, P.: An FPGA synthesis of the distributed control systems designed with Petri nets. In: Proc. IEEE 3rd Int. Conf. on Networked Embedded Systems for Every Application, Liverpool, UK (2012) [6]
4. Karatkevich, A., Wiśniewski, R.: Computation of Petri nets covering by SM-components based on the graph theory. Electrical Review (August 2012) 141-144
5. Rudell, R.L.: Logic Synthesis for VLSI Design. PhD thesis, University of California (1989)
6. Wiśniewski, R., Wiśniewska, M., Adamski, M.: A polynomial algorithm to compute the concurrency hypergraph in Petri nets. Measurement Automation and Monitoring 58(7) (2012) 650-652 in Polish.
7. Wiśniewska, M.: Application of Hypergraphs in Decomposition of Discrete Systems. Volume 23 of LNCCS. University of Zielona Góra Press, Zielona Góra (2012)
8. Stefanowicz, L., Adamski, M., Wiśniewski, R.: Application of an Exact Transversal Hypergraph in Selection of SM-Components. IFIP Advanced in Information and Communication Technology (2013)
9. Knuth, D.: Dancing links. Millennial Perspectives in Computer Science (2000)
10. Eiter, T.: Exact transversal hypergraphs and application to boolean u-functions. Journal of Symbolic Computation 17(3) (1994) 215-225
11. Kovalyov, A.: Concurrency Relations and the Safety Problem for Petri Nets. Application and Theory of Petri Nets (1992)
12. Berge, C.: Hypergraphs: Combinatorics of Finite Sets. North-Holland (1989)
13. Bukowiec, A., Doligalski M.: Petri Net Dynamic Partial Reconfiguration in FPGA. Computer Aided Systems Theory – EUROCAST (2013) 436-443
14. Karatkevich, A.: SM-Components problem reductions of Petri nets. Telecommunication Review (2008) in Polish.
15. Barkalov, A., Titarenko, L., Bieganowski, J., Miroshkin, A.: Synthesis of Compositional Microprogram Control Unit with Dedicated Area of Inputs. In: Lecture Notes in Electrical Engineering. Number 79 (2011) 193-214
16. Barkalov, A., Kołopieńczyk, M., Titarenko, L.: Design of CMCU with EOLC and encoding of collections of microoperations. (79) (2007) 262-265
17. Blanchard, M.: Comprendre, maitriser et appliquer le Grafcet. Automatisation Production (1979)
18. DeMicheli, G.: Synthesis and Optimization of Digital Circuits. PhD thesis, McGraw-Hill Higher Education (1994)
19. Bukowiec, A., Barkalov A.: Automata Implementation in FPGA devices with Multiple Encoding States. Electrical Review (2009) 185-188